

## PRIMER LABORATORIO

### “ADMINISTRACIÓN BÁSICA DE UN SGBDR”

#### CONSIGNAS GENERALES:

- La modalidad de esta instancia de evaluación es de carácter grupal y la nota obtenida se promedia con los otros laboratorios. Se tendrá en cuenta una nota conceptual en base a la organización grupal y participación de cada integrante, por lo que las notas individuales dentro del grupo pueden diferir.
- Para la realización del laboratorio el grupo deberá utilizar la máquina virtual PROPORCIONADA POR LA CÁTEDRA para las clases de práctica.
- Puede utilizarse como material de consulta la documentación oficial del SGBD.
- Deberán responder los ejercicios subiendo luego a la actividad del laboratorio en el aula virtual un *documento con los enunciados, las sentencias, instrucciones o comandos ejecutados y con una captura de pantalla con los resultados de cada ejecución y/o una imagen que permita comprobar que la solución obtenida resuelve el problema propuesto*. Si se considera necesario, al final del ejercicio se puede agregar nota aclaratoria o especificando algún inconveniente encontrado.
- **TENER EN CUENTA EL TIEMPO LÍMITE PARA FINALIZAR EL LABORATORIO (DISPONIBLE EN LA ACTIVIDAD EN EL AULA VIRTUAL).**

#### EJERCICIOS:

- 1) **(5p) Crear en el SGBD una cuenta de usuario** con el nombre del grupo (p.e. “usr\_g1”) y con acceso desde cualquier host. Otorgar los permisos necesarios teniendo en cuenta que *exclusivamente con esta cuenta* se deben realizar el resto de los ejercicios de la guía, es decir que **no se debe usar la cuenta “root” u otra que se haya creado anteriormente**.
- 2) **(15p)** Descargar e instalar la BD de ejemplo “employees” de acuerdo a las instrucciones en la documentación oficial de MySQL (<https://dev.mysql.com/doc/employee/en/employees-installation.html>). Si ya se dispone de la base de datos “employees” en el servidor, eliminarla previamente.
  1. **Crear una cuenta de usuario** con acceso local que tenga los *privilegio mínimos necesarios* para realizar copias de respaldo de la base de datos recién instalada (escribir como comentario la justificación de cada privilegio otorgado). Con esta nueva cuenta:
    1. **Realizar una copia de seguridad** solamente de los datos almacenados en las tablas “employees”, “salaries” y “titles”. Considerar la realización de estos respaldos en *formato de texto delimitado* (cap. 6.5.4 y 9.4.3 del manual de referencia), teniendo en

- cuenta la configuración de variables del sistema, directorios de archivos y posibles permisos necesarios en el sistema operativo.
2. Crear una base de datos de nombre **“empleados”** con tres tablas: **“empleados”**, **“salarios”** y **“cargos”**. Los esquemas de estas tablas deberán tener la misma estructura que **“employees”**, **“salaries”** y **“titles”** en la base de datos **“employees”** pero con los nombres de los campos en castellano.
  3. **Hacer una carga masiva** de los datos respaldados en el ítem 2)1.1 en las tablas **“empleados”**, **“salarios”** y **“cargos”**.
- 3) **(25p)** Para el siguiente ejercicio utilizar la base de datos **“employees”**. Se puede tomar como referencia el tutorial disponible en **“Indices y Analizador Sintactico.pdf”**. Considerar como apoyo el esquema de estructura de tablas disponible en la documentación de la base de datos **“Employees”**. Luego, para cada una de las consultas planteadas a continuación **registrar**:
- Cantidad de **registros devueltos**;
  - **Tiempos** de ejecución;
  - **Información devuelta por la sentencia EXPLAIN** (cantidad de filas, operaciones, índices, costos, etc.), ya sea en forma textual (en el formato de salida que consideren adecuado) o gráfica. Considerar el uso del modificador **ANALYZE**.
1. Listar los datos completos de todos los empleados con su respectivo historial de salarios;
  2. Al listado anterior agregar una condición (en WHERE) sobre el apellido (last\_name), por ej. **“LIKE ‘Gut%’** o **“= ‘González’”**. Prever que la consulta devuelva más de un registro y esté ordenado por apellido.
  3. Al listado en el ejercicio 1. agregar una condición (en WHERE) para que el monto del salario (salary) cumpla algún criterio de igualdad o rango, por ejemplo **“= 55555”** o **“BETWEEN 33333 AND 44444”** o **“> 70000 AND < 80000”**. Prever que la consulta devuelva registros actuales (campos **“to\_date”** = ‘9999-01-01’).
  4. Al listado en 1. agregarle conjuntamente las condiciones de 2. y 3. Esta consulta realizarla de manera distinta a las anteriores (p.e. distintos operadores, uso de CTE/TDL, etc.).
  5. Crear un índice por **“last\_name”** en **“employees”** y ejecutar nuevamente las consultas de 1. a 4.
  6. Crear un índice por **“salary”** en **“salaries”** y ejecutar nuevamente las consultas de 1. a 4.
  7. Teniendo en cuenta que hay distintos cargos de Ingenieros (**Engineer**, **Senior Engineer** y **Assistant Engineer** en la tabla **titles**), listar todos los datos de los empleados con el historial de salarios y cargos ocupados, pero sólo para los cargos de Ingenieros. Hacer esta consulta evaluando el campo **title** una vez con **“LIKE”** (para considerar todos los cargos de una vez) y otra con **“=”** (considerando los distintos cargos por separado).
  8. Escribir un breve resumen comparativo (no más de 15 renglones) como conclusión de los datos registrados en cada ejecución.

- 4) (20) En este ejercicio, trabajar con la base de datos “Employees”. Tener en cuenta que los campos “*to\_date*” registran el valor ‘9999-01-01’ para reflejar **datos actuales** (sueldo actual, cargo actual, etc.). Utilice las técnicas/herramientas que considere adecuadas para resolver cada situación planteada, explicando por qué lo hace de esa manera.
1. Dado el número de empleado de un administrador (“emp\_no” en tabla “dept\_manager”), se necesita mostrar el nombre del departamento que administra y los datos de los **empleados que trabajan actualmente** en él: *nombre completo, cargo actual, salario actual y antigüedad* (“hire\_date” es la fecha de contratación). Verificar previamente que el número de empleado dado es quien administra actualmente un departamento, y en caso contrario mostrar una leyenda que notifique que el empleado no administra actualmente un departamento.
  2. Cuando un empleado es dado de alta por primera vez como gerente de un departamento (tabla “dept\_manager”) se debe cerrar su periodo actual si figura como trabajador (tabla “dept\_emp”) y cerrar también su cargo actual (tabla “titles”) y darlo de alta con la fecha actual y con el cargo de “**Manager**”.
  3. En la empresa se decide empezar a registrar a los/as hijos/as de los empleados (nombre completo, fecha de nacimiento) para poder asignarles una ayuda escolar a aquellos que tengan menos de 18 años.
- 5) (20) Utilizando la base de datos **employees** simular una situación en la que:
1. desde una sesión, se debe bloquear una tabla para realizar una transacción (p.e. aumentar en un porcentaje dado el salario, en la tabla *salaries*, de los empleados con determinado cargo);
  2. paralelamente, desde otra sesión se requiere acceder a la misma tabla para realizar otra operación de actualización o borrado y ésta queda bloqueada a la espera de que finalice la primera;
  3. en ese momento ejecute las sentencias adecuadas para que el SGBD muestre la siguiente información: *identificador del del hilo de ejecución y de la transacción que está bloqueando y de la que está esperando, consulta que está bloqueando y consulta que está esperando*;
  4. por último “mate” el hilo de ejecución de la consulta iniciada en segundo término.
  5. **Antes de empezar**, considerar lo siguiente:
    - Crear una cuenta de usuario distinto que pueda realizar operaciones de lectura y escritura en la base de datos employees, y realizar con ésta una de las sesiones para diferenciarlas;
    - Se puede usar como guía el tutorial disponible en “**Tutorial bloqueos.pdf**”.
    - Analizar el uso de las sentencias “SHOW PROCESSLIST” y “KILL” y de la “información sobre las transacciones y bloqueos InnoDB” en las secciones 15.7.7.31, 15.7.8.4 y 17.15.2.1 del manual de referencia.

- 6) **(15p)** Para este ejercicio utilizar la base de datos “sakila”. Asegurarse de que se están ejecutando respaldos incrementales mediante binary logs (hacer una captura de pantalla del listado de los actuales) y realizar las siguientes acciones:
1. Crear un evento que cada 10 minutos realice un vaciado (flush) de los logs binarios y guarde en una tabla el momento en que se realiza esta acción (esta tabla debe crearse antes de programar el evento).
  2. **Crear un respaldo completo** de la base de datos “sakila” sobre la que se hicieron los ejercicios de las guías prácticas asegurándose de contemplar la disponibilidad de respaldos incrementales.
  3. Insertar y/o actualizar al menos 5 registros en distintas tablas que puedan ser fácilmente identificables después.
  4. **Crear una cuenta de usuario** con permiso de acceso desde hosts externos y darle todos los privilegios para la base de datos “sakila”.
  5. **Conectarse al servidor con MySQL Workbench** utilizando la cuenta creada en el ítem anterior.
  6. Confeccionar y ejecutar 5 consultas que el grupo considere “interesantes” (de acuerdo a lo practicado como “consultas SQL avanzadas) mediante la conexión anterior y registrar sus resultados.
  7. “Romper” la base de datos accedida.
  8. Repetir el ítem 4) 3. (inserción/actualización de datos) con otros registros.
  9. **Restaurar** lo que se rompió, asegurándose de recuperar los datos de 4) 3. y 4) 8..
  10. Terminar el laboratorio con una captura de pantalla de los binary logs actuales.